Comparison of Sudoku Solving Algorithms Calum Harvey

1. Motivation and Rationale

1.1 The Context:

Sudoku is a logic-based number placement puzzle game. The aim is to fill a grid, usually 9x9, with numbers so that each row, each column and each smaller 3x3 box contains the digits from 1-9 only once. When a puzzle is created a portion of the grid is already completed and a good solution will allow the solver to come to only one solution.

1.2 The Problem:

When a sudoku is solved by a human they use logic by scanning rows and columns in each 3-box area finding places where only a single number can fit in a single square but for a computer to simulate this is almost impossible. Therefore, three main types of algorithm have been developed for solving this problem: Backtracking, stochastic and constraint satisfaction problem. The problem is that these algorithm types perform differently with backtracking being a brute-force method of solving meaning it always gets the result but not necessarily the fastest with the other two being more general algorithm types that are “smarter” and try and make meaningful progress based on the state of the puzzle. These need to be compared to find the best algorithm at different complexities of puzzle based on time taken and iterations of algorithm.

1.3 The Rationale:

In my project, the intent of the project is to develop a test bed that allows for sudoku puzzles to be created at different levels of complexity that will allow me to compare algorithms from the three different categories of potential sudoku solving algorithms. With a standard implementation of backtracking and multiple algorithms from the stochastic category of algorithm that I find in papers. This will allow me to compare the implemented algorithms using different puzzles including puzzles that have no single solution.

2. Aims & Objectives

2.1 Aim: To explore backtracking, stochastic and constraint satisfaction problem sudoku solving algorithms and compare and contrast them using a range of difficulty of puzzles.

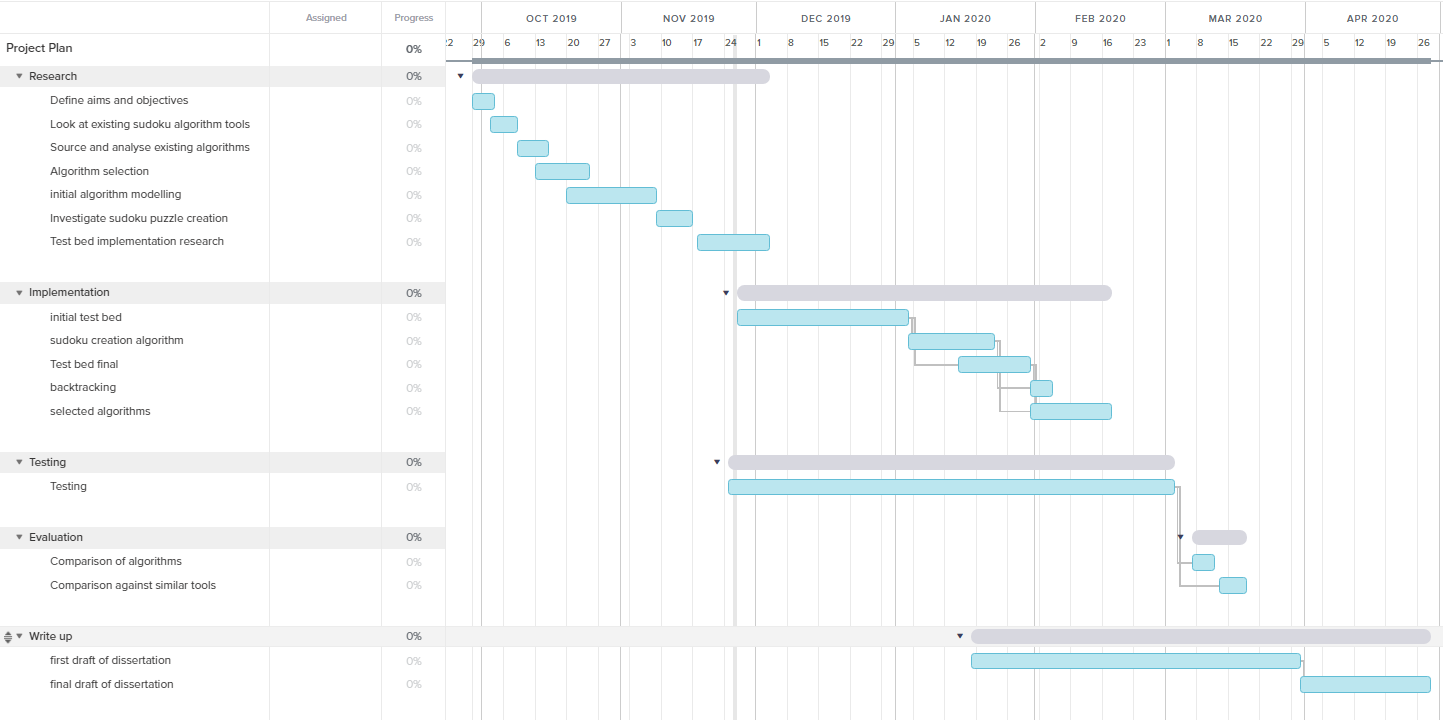
2.2 Objectives:

|  |  |
| --- | --- |
| Objectives | Explanation |
| *1. Explore current methods of sudoku solving and select three* | In my background reading, I will research the current algorithms in each of the three types of algorithms for sudoku solving. After my research is complete, I will select three algorithms that I will be able to implement. These need to be sufficiently different to make a good comparison but also feasible within the timeframe of the project. |

|  |  |
| --- | --- |
| *2. Develop test bed to allow comparison of algorithms* | The test bed will be developed to allow all the other elements of the project to be used by each other. This is important as there needs to be a way for the test data to be given to the algorithms. This test bed will allow each of the algorithms to be tested against an array of puzzles from easy to impossible. |
| *3. Establish test data for comparison of algorithms* | When comparing the algorithms, having test data that can be controlled and created without bias is important. I will research and develop sudoku creation algorithms that will allow me to create test data at specific difficulties, providing a fair comparison of the algorithms at different levels of puzzle. |
| *4. Implement algorithms into the test bed* | The algorithms that I will implement are the three that I chose during original research into algorithms. The algorithms need to be implemented last as they need the test data and test bed to be able to be tested that they work correctly. Another algorithm that will be implemented is a backtracking algorithm as it provides an example of brute force against the more efficient algorithms. |
| *5. Evaluate implemented sudoku algorithms at multiple complexities* | The evaluation will be done by using time taken for the algorithm to solve the sudoku and also the number of iterations each one takes. This allows two different criteria to be compared, which one is faster but also which one is more efficient which could allow for an interesting comparison. |
| *6. Explore state of the art sudoku solving tools and evaluate selection* | I want a broader evaluation of my system as a whole against other tools that are already available for evaluating and comparing sudoku solving algorithms. This will allow me to understand how successful my project was and the satisfaction of this objective dictates the success of the project. |

3. Background

|  |  |  |
| --- | --- | --- |
| Resource | Description | Relevance |
| *3.1 [1]* | This paper looks at stochastic algorithms for solving sudoku puzzles and also introduces a new stochastic method for solving these puzzles. It then tests the new algorithm for its performance under different conditions of puzzle. | As my plan is to pick three stochastic algorithms to compare, this paper creating an stochastic algorithm is useful as I could use this algorithm as one of the algorithms to compare. |
| *3.2 [2]* | This paper discusses four stochastic search techniques that it then implements them and compares using time taken to solve the puzzle and the number of iterations. | This is relevant to my project as it takes sudoku solving algorithms and compares them so I can get an idea of what I need to do during my project. This paper also shows more types of stochastic algorithm that I could use. |
| *3.3 [3]* | It describes backtracking and recursion that is used for sudoku solving brute force. It also gives pseudocode on implementing backtracking in sudoku solving. | I plan on implementing backtracking as a side task of my project and the pseudo code and example of how to implement backtracking will help me to implement this into my project faster. |
| *3.4 [4]* | This paper examines the possibility of using genetic algorithms to solve sudoku and FPGA implementation to solve the puzzle. | This is a paper focused on genetic algorithms which are a type of stochastic algorithm that I will look to implement and the insight into how this type of algorithm will perform will help during my development. |
| *3.5 [5]* | This paper discusses the methods of creating sudoku puzzles and how the difficulty of such puzzles can be defined. | I need to create sudoku puzzles in order to test my algorithms that I am implementing and need to be able to alter difficulty of these puzzles and this paper describes a way for me to do this. |

4. Work Plan

5. Explanation of work plan

To start my project, I began with some background research into similar systems that already exist to see how they could be improved and also into current sudoku solving algorithms to find out what research has already been done into them individually and trying to understand the complexity of them. Using this research, I then selected the three algorithms that I am going to implement and modelled them, this meets my first objective.

I then began research into implementing the test bed and will begin implementing this before Christmas, as this is the base of the software that everything else that will be implemented will be attached to. Therefore, this needs to be finished early to allow me to complete other parts of the project and to test them effectively. This satisfies my second objective.

After I have finished the initial test bed, I will begin development on the sudoku creation which is the test data that I will use to compare the algorithms. Using my previous research into the area, this algorithm should be fairly straight forward to implement due to the amount of papers that have been written on the subject. Once the sudoku creation is complete my third objective is complete.

During the implementation of the test data algorithm, I will also finalise the test bed as I believe I will find ways to improve the initial implementation during the development of the sudoku creation algorithm as it is being added to the test bed. After this I will move on to implementing the algorithms. The first one implemented is backtracking as it is a simple algorithm to implement and provides a proof of concept that the rest of the algorithms will work when implemented into the testbed. When the rest of the solving algorithms have been implemented, the fourth objective will be satisfied.

Although testing will be done throughout the development, I have given myself time after I have finished all the development to either catch up if I have fallen behind and for testing of the system as a whole. This testing is important as I need to make sure the results of the comparisons between the algorithms is accurate and there aren’t any bugs causing the data to be wrong. I will then compare the algorithms based on time taken to solve a range of complexity of puzzles and also the number of iterations that each one takes as that will show me not only how fast each algorithm is but also how efficient it is, completing the fifth objective.

The final objective will be completed by using the research of similar systems to compare my final project with those other tools to see if I improved on them and giving me an insight into the overall success of the project.

The first draft of the dissertation will be written over the whole development phase as it is important to write the relevant sections when it is fresh in my memory. I will then get feedback on my first draft and make improvements on it during the Easter holidays before the submission.

The main risk I have during this project is the amount of time I have. This is due to each stage in the development being dependent on the stage before and therefore, if I take more time on an earlier section this will cause the later sections to also be delayed. I have solved this in my plan by adding extra time to my testing section that will allow to catch up if I have fallen behind on the work as well as test the final project.

6. References

[1] Lewis, R. (2007). Metaheuristics can solve sudoku puzzles. *Journal of Heuristics*, 13(4), pp.387–401.

[2] Perez, M. and Marwala, T. (2008). STOCHASTIC OPTIMIZATION APPROACHES FOR SOLVING SUDOKU.

[3] Zelenski, J. (2008). Exhaustive recursion and backtracking.

[4] Thirer, N. (2012). About the FPGA implementation of a genetic algorithm for solving Sudoku puzzles. 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel.

[5] Meng, J. and Lu, X. (2011). The Design of the Algorithm of Creating Sudoku Puzzle. *Lecture Notes in Computer Science*, pp.427–433.

‌